

REMARKS

This is a full and timely response to the non-final Official Action mailed **September 19, 2008** (the “Office Action” or “Action”). Reconsideration of the application in light of the above amendments and the following remarks is respectfully requested.

Claim Status:

By the foregoing amendment, claims 15 and 26 have been amended, and claim 24 has been cancelled without prejudice or disclaimer. Claims 5, 16-23 and 25 were cancelled without prejudice or disclaimer by a previous paper. No new claims have been added by the present paper. Thus, claims 1-4, 6-15 and 26 are currently pending for further action.

Objection to Claims:

In the outstanding Office Action, the Examiner objected to claim 15 because of an informality. This issue has been corrected by the present amendment. No new matter has been added. Moreover, the amendments made in this regard do not, and are not intended to, narrow or change the scope of the claims. Following entry of this amendment, the objection to the claims may be reconsidered and withdrawn.

35 U.S.C. § 101:

In the recent Office Action, claims 24 and 26 were rejected under 35 U.S.C. § 101. This rejection is rendered moot by the cancellation of claim 24 and the amendment of claim 26 to

depend from claim 15. Following this amendment, all the remaining claims are believed to be in compliance with 35 U.S.C. § 101 and notice to that effect is respectfully requested.

Prior Art:

With regard to the prior art, claims 1-3, 15, 24 and 26 were rejected under 35 U.S.C. § 102(b) as being anticipated by Frank L. Friedman & Elliot B. Koffman, *Problem Solving, Abstraction, and Design Using C++* (2d ed. 1997), (“Friedman”). For at least the following reasons, this rejection is respectfully traversed.

Claim 1 recites:

A method of executing a computer algorithm, comprising:
executing a first module encapsulating said computer algorithm except at least one communication operation of said algorithm;
executing a second module encapsulating said at least one communication operation of said algorithm, such that said at least one communication operation is available to said first module; and
instantiating at least one data object for encapsulating data communicated between said first module and a communicating partner, each one of said at least one data object being an instance of a data class, said data communicated between said first module and said communicating partner being accessible by said first module.

(Emphasis added).

In contrast, Friedman is directed to an elementary discussion of the principles of object oriented programming using the C++ programming language. As such, Friedman teaches general theories relating to the nature of algorithms and data objects and the practical application of those theories in solving basic programming problems. However, nowhere does Friedman teach or suggest the idea of separate modules encapsulating different portions of an algorithm.

According to the recent Office Action, Friedman’s discussion of an exemplary Circle class and member functions of the Circle class read on the “first module encapsulating said

computer algorithm except at least one communication operation of said algorithm” recited by claim 1. (Action, p. 3; *See also* Friedman, pp. 514-15 and Fig. 11.8). This assertion is clearly incorrect. In the first place, it is elementary in object-oriented programming that a data class is a form of abstraction in which the attributes and functions of a type of data object are defined. Thus, a data class is not an algorithm, nor is a data class analogous to an algorithm. Rather, a data class defines the organization (i.e. abstraction) of data that can be passed to and manipulated by an object-oriented algorithm. Friedman makes the distinction very clear by defining an algorithm as “a list of steps,” and pointing out that the illustrative Circle class is “used to represent circle objects.” (Friedman, pp. 21 and 510). The terms used in Friedman must be construed and interpreted in light of Friedman’s own definitions and their accepted usage in the art, neither of which accords with the Examiner’s interpretation of these terms.

As was briefly mentioned above, a data class can define functions that may be performed on an object that instantiates the data class. It appears that the functions defined in Friedman’s Circle class may be one source of confusion for the Examiner in misconstruing the Circle class as an algorithm. Applicant wishes to point out the well-established object-oriented programming principle that a data class merely sets forth *various* functions that *may* be performed on a data object type while an algorithm sets forth *which* functions *will* be performed on the object, the ordering and conditions under which those functions will be performed on the object, and the data parameters that will be used in conjunction with the functions. By way of analogy, the functions defined in a data class set forth the tools that may be used on a data object of that class while an algorithm sets forth a plan of how those tools will be used on specific data objects to

accomplish a particular task. The tools (functions defined in a class) should not to be mistaken for the plan (algorithm) of how those tools are to be used.

The Office Action perpetuates this misunderstanding of the difference between class functions and algorithms by further asserting that Friedman’s “driver function to test class circle” reads on the “second module encapsulating said at least one communication operation of said algorithm” recited by claim 1. (Action, p. 3; *See also* Friedman, p. 513). It will be readily apparent to anyone having even cursory experience in computer programming experience that the “driver function to test class circle” taught by Friedman is an entire algorithm in and of itself. The driver function is a very simple algorithm in that it merely instantiates an object of the Circle class and tests the functions defined in the Circle class on the newly created data object, but nonetheless it is a complete algorithm. Nowhere does Friedman teach or suggest that the “driver function to test class circle” algorithm is modularized into different encapsulations of functionality, with “a first module encapsulating said computer algorithm except at least one communicating operation of said algorithm” and “a second module encapsulating said at least one communication operation of said algorithm, such that said at least one communication operation is available to said first module.” (Friedman, p. 513; claim 1). Friedman simply does not teach or suggest this subject matter.

“A claim is anticipated [under 35 U.S.C. § 102] only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987) (emphasis added). See M.P.E.P. § 2131. For at least these reasons, the rejection based on Friedman of claim 1 and its dependent claims should be reconsidered and withdrawn.

Claim 15 now recites:

A computer readable medium storing thereon computer executable instruction code, said code when executed by a processor of a computer causes said processor to:
execute a first module encapsulating a computer algorithm except at least one communication operation of said algorithm; and

execute a second module encapsulating said at least one communication operation of said algorithm, such that said at least one communication operation is available to said first module,

wherein said second module encapsulates at least one environment-dependent communication operation of said algorithm and is configured to communicate with a communicating partner.

(Emphasis added).

In contrast, Friedman does not teach or suggest the subject matter of claim 15.

Specifically, as has been amply demonstrated above, Friedman fails to teach or suggest “a first module encapsulating a computer algorithm except at least one communication operation of said algorithm” and “a second module encapsulating said at least one communication operation of said algorithm, such that said at least one communication operation is available to said first module.” (claim 15). As Friedman does not teach or suggest the “second module” recited in claim 15, Friedman *cannot* teach or disclose that “said second module encapsulates at least one environment-dependent communication operation of said algorithm,” also recited in claim 15.

“A claim is anticipated [under 35 U.S.C. § 102] only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987) (emphasis added). See M.P.E.P. § 2131. For at least these reasons, the rejection based on Friedman of claim 15 and its dependent claims should be reconsidered and withdrawn.

Claims 4 and 6-14 were rejected under 35 U.S.C. § 103(a) as being obvious in light of the teachings of Friedman taken alone. This rejection is respectfully traversed for at least the reasons given above in favor of the patentability of independent claim 1.

Additionally, various dependent claims of the application recite subject matter that is further patentable over the cited prior art. Specific, non-exclusive examples follow.

Claim 2:

Claim 2 recites “wherein said at least one communication operation comprises at least one environment-dependent communication operation of said algorithm.” In response, the Office Action cites again to Friedman’s “driver function to test class circle” and the Circle class itself, stating that “if changes need to be made to driver function, circle class would not be affected.” (Action, p. 4; *see also* Friedman, p. 513). While this statement is true, it is moot in light of the fact that neither the Circle class nor the driver function to test the Circle class can be reasonably construed as a module encapsulating a portion of an algorithm. Moreover, Friedman does not appear to teach or suggest this subject matter anywhere else. For at least this additional reason, the rejection of claim 2 should be reconsidered and withdrawn.

Claim 4:

Claim 4 recites “executing a third module encapsulating another communication operation of said algorithm.” As has been amply demonstrated above, Friedman does not teach or suggest first and second modules encapsulating different portions of an algorithm. Therefore,

Friedman *cannot* teach or suggest the third module encapsulating another communication operation of said algorithm. For at least this additional reason, the rejection of claim 4 should be reconsidered and withdrawn.

Claims 13-14:

Claim 13 recites “wherein said at least one method of communication comprises a method of communicating data from said communicating partner to said second data object” and claim 14 recites “wherein said at least one method of communication comprises a method of communicating data from said communicating partner to said second data object.” Friedman does not teach or suggest this subject matter. For at least this additional reason, the rejection of claims 13-14 should be reconsidered and withdrawn.

Conclusion:

In view of the foregoing arguments, all claims are believed to be in condition for allowance over the prior art of record. Therefore, this response is believed to be a complete response to the Office Action. However, Applicant reserves the right to set forth further arguments in future papers supporting the patentability of any of the claims, including the separate patentability of the dependent claims not explicitly addressed herein. In addition, because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed.

The absence of a reply to a specific rejection, issue or comment in the Office Action does not signify agreement with or concession of that rejection, issue or comment. Finally, nothing

in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment. Further, for any instances in which the Examiner took Official Notice in the Office Action, Applicants expressly do not acquiesce to the taking of Official Notice, and respectfully request that the Examiner provide an affidavit to support the Official Notice taken in the next Office Action, as required by 37 CFR 1.104(d)(2) and MPEP § 2144.03.

If the Examiner has any comments or suggestions which could place this application in better form, the Examiner is requested to telephone the undersigned attorney at the number listed below.

Respectfully submitted,

DATE: October 31, 2008

/Steven L. Nichols/
Steven L. Nichols
Registration No. 40,326

Steven L. Nichols, Esq.
Managing Partner, Utah Office
Rader Fishman & Grauer PLLC
River Park Corporate Center One
10653 S. River Front Parkway, Suite 150
South Jordan, Utah 84095

(801) 572-8066
(801) 572-7666 (fax)